

IN THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A method, comprising:
interacting with a debugging interface for receiving instructions to debug during a debugging session;
maintaining coherence between a simulation mode and an emulation mode by maintaining mappings between physical and logical resources during the debugging session; and
dynamically switching between the emulation mode and the simulation mode during the debugging session by stopping a current mode of operation during the session and accessing the mappings to switch to a new mode of operation during the session, wherein the debugging session interactively debugs the instructions and can be switched between the emulation mode and the simulation mode dynamically on user instruction or automated instruction.
2. (Original) The method of claim 1 further comprising debugging a number of the instructions by a simulator while in the simulation mode.
3. (Original) The method of claim 1 further comprising debugging a number of the instructions by an emulator while in the emulation mode.
4. (Original) The method of claim 1 wherein interacting further includes displaying, by the debugging interface, a current state of a resource associated with the instructions during the debugging session.
5. (Original) The method of claim 1 wherein interacting further includes receiving, by the debugging interface, manual commands to debug the instructions during the debugging session.
6. (Original) The method of claim 1 wherein interacting further includes receiving, by the debugging interface, script commands to debug the instructions during the debugging session.

7. (Original) The method of claim 6 wherein interacting further includes receiving by the debugging interface, a switch command to process the switching between the emulation mode and the simulation mode.

8. (Currently Amended) A method, comprising:
establishing a debugging session to debug instructions, wherein the debugging session is interactive;

maintaining coherent states between logical and physical resources that are used to debug the instructions during the debugging session by maintaining mappings between physical and logical resources used during the debug session; and

dynamically passing control of the debugging session between a simulator that manages the logical resources and an emulator that manages the physical resources during the debugging session and while the instructions are still being executed and dynamically passing control by stopping the simulator and using the mappings to transition to the physical resources and initiating the emulator during the debugging session.

9. (Original) The method of claim 8 further comprising interfacing with a debugging interface that supplies a switch mode command, which drives the passing of control.

10. (Original) The method of claim 9 further comprising transmitting the coherent states to the debugging interface.

11. (Original) The method of claim 8 further comprising interfacing with a script that supplies a switch mode command, which drives the passing of control.

12. (Original) The method of claim 8 further comprising:
acquiring control from the simulator when control is with the simulator in response to receiving a switch mode command; and
passing control to the emulator.

13. (Original) The method of claim 8 further comprising:
acquiring control from the emulator when control is with the emulator in response to receiving a switch mode command; and
passing control to the simulator.
14. (Original) The method of claim 8 further comprising, receiving the instructions from a debugging interface.
15. (Currently Amended) A system, comprising:
a debugging interface to receive instructions that are to be debugged during a debugging session;
a controlling interface to maintain coherence between a simulation mode and an emulation mode associated with the debugging session, wherein coherence is achieved via mappings maintained from logical resources associated with the simulation mode and physical resources associated with the emulation mode; and
a debugging execution interface to dynamically debug a number of the instructions while the debugging session is in the simulation mode or the emulation mode;
wherein the debugging interface communicates with the controlling interface and the controlling interface communicates with the debugging execution interface, and wherein the debugging interface permits dynamic switching between the simulation and emulation modes while the instructions are being debugged by stopping a current mode of operation during the session and using the mappings to dynamically switch to a new mode of operation during the session.
16. (Original) The system of claim 15 wherein the debugging execution interface further includes:
a simulator to process a number of the instructions while in the simulation mode; and
an emulator to process a number of the instructions while in the emulation mode.

17. (Original) The system of claim 15 wherein the debugging interface supplies the instructions to the controlling interface.
18. (Original) The system of claim 15 wherein the debugging interface supplies a switch mode command to the controlling interface and in response to the switch mode command the controlling interface sets at least one of the simulation mode and emulation mode and switches control of the debugging session by communicating with the debugging execution interface.
19. (Currently Amended) A machine accessible medium having associated data, which when accessed, carries out in a machine the method of:
- establishing a debugging session to debug instructions, wherein the debugging session is interactive;
 - maintaining coherent states between logical and physical resources that are used to debug the instructions during the debugging session by maintaining mappings between the logical and the physical resources; and
 - dynamically changing control between simulation modes associated with the logical resources and emulation modes associated with the physical resources during the debugging session and while the instructions are being debugged by stopping processing associated with the simulation modes and accessing the mappings to initialize and start the emulations modes.
20. (Original) The medium of claim 19 wherein the debugging session is established and driven by a debugging interface.
21. (Original) The medium of claim 19 wherein control is changed between a simulator for the simulation modes and an emulator for the emulation modes.
22. (Currently Amended) An apparatus in a computer accessible medium comprising:
- a simulator;
 - an emulator; and

a debugging session manager, wherein the debugging session manager dynamically manages a debugging session to debug instructions by maintaining mappings of logical resources to physical resources, and wherein during the debugging session a number of the instructions are selectively processed by the emulator and a number of the instructions are selectively processed by the simulator, and wherein switching from the simulator to the emulator is achieved while the instructions are being debugged by stopping a current mode of operation and accessing the mappings to switch to a new mode of operation during the debugging session.

23. (Original) The apparatus of claim 22 wherein the debugging session manager interacts with a debugging interface to receive the debug instructions and establish the debugging session.
24. (Original) The apparatus of claim 23 wherein the debugging session manager selectively determines which of the instructions that the simulator and emulator process based on commands received from the debugging interface.
25. (Original) The apparatus of claim 23, wherein the debug session manager passes control between the simulator and the emulator as many times as is requested during the debug session.